UNCLASSIFIED

| AD NUMBER |
|---|
| **ADB157201** |
| LIMITATION CHANGES |
| TO:<br>**Approved for public release; distribution is unlimited.** |
| FROM:<br>**Distribution authorized to U.S. Gov't. agencies and their contractors; Specific Authority; 04 AUG 1988. Other requests shall be referred to Air Force Electronic Systems Division, ESD/AVS, Hanscom AFB, MA, 01731-5000.** |
| AUTHORITY |
| **ESD per IBM ltr dtd 11 Oct 1991** |

THIS PAGE IS UNCLASSIFIED

Task/Subtask IS15.2
CDRL Sequence 03705-001B
30 September 1991

2

# SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS) PROGRAM

## SPMS Training Class:
## Student Handout

## Addendum to:
## Software Process Tools & Techniques Report

Contract No. F19628-88-D-0032

Task IS15 - Software Process Management

Prepared for:

Electronic Systems Division
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5000

Prepared by:

IBM Federal Sector Division
800 North Frederick Avenue
Gaithersburg, MD 20879

Specific Authority

Distribution Authorized to U.S. Government Agencies and Their Contractors; ~~Pursuant to Data Attachment in Contract F19628-88-D-0032~~, Dated 04 August 1988.

Other Requests for this Document Must Be Referred to ESD/AVS, Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, MA 01731-5000.

Task/Subtask IS15.2
CDRL Sequence 03705-001B
30 September 1991

# SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS) PROGRAM

## SPMS Training Class: Student Handout

## Addendum to: Software Process Tools & Techniques Report

Contract No. F19628-88-D-0032

Task IS15 - Software Process Management

Prepared for:

Electronic Systems Division
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5000

Prepared by:

IBM Federal Sector Division
800 North Frederick Avenue
Gaithersburg, MD 20879

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
September 30, 1991

3. REPORT TYPE AND DATES COVERED
Initial

4. TITLE AND SUBTITLE
Software Process Management System Student Handout

5. FUNDING NUMBERS
F19628-88-D-0032/0005

6. AUTHOR(S)
William H. Ett, IBM

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
IBM Federal Sector Division
800 North Frederick Avenue
Gaithersburg, MD 20879

8. PERFORMING ORGANIZATION REPORT NUMBER
03705-001B

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Electronic Systems Division/AVK
Air Force Systems Command, USAF
Hanscom Air Force Base, MA 01731-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
N/A

12a. DISTRIBUTION/AVAILABILITY STATEMENT
~~Limited to U.S. Government Agencies and their Contractors~~

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

This document is the student handout prepared for the "SEI/STARS P.3 Asset Acquisition Sub-task" training class. The student handout covers basic aspects of process management with respect to the "Software Process Management System". The handout also includes several sample workshop problems.

This document is a companion document to IBM CDRL 03705-001, the "Software Process Tools and Techniques Evaluation Report".

14. SUBJECT TERMS
Software Process Management, Software Process Modeling, Training Materials

15. NUMBER OF PAGES

16. PRICE CODE
N/A

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
SAR

This document was developed by the IBM Federal Sector Division, located at 800 North Frederick Avenue, Gaithersburg, MD 20879. Questions or comments should be directed to the document owner William H. Ett, IBM, (Internet: ETTB@WMAVM7.iinus1.ibm.com).

This document is approved for release under *Distribution "C"* of the *Scientific and Technical Information Program Classification Scheme* (DoD Directive 5230.24). Permission to use, modify, copy or comment on this document for purposes stated under Distribution "C" without fee is is hereby granted. The Government (IBM and its subcontractors) disclaims all responsibility against liability, including expenses for violation of proprietary rights, or copyrights arising out use of this document. In addition, the Government (IBM and its subcontractors) disclaims all warranties with regard to this document. In no event shall the Government (IBM nor its subcontractors) be liable for any damages in connection with the use of this document.

## Training Materials for the

# SPMS

## Software Process Management System

Click Anywhere To Begin

Another great tool from the SPI Group

# Course Objectives

The course will focus on :

- Using the Software Process Management System
- Learning to create process models
- Learning how each element of the SPMS interacts with other elements
- Learning how to use the project specific plans to assist in validation of the process models

By the end of the course , students will have:

- Become familiar with the elements of the SPMS
- Created, instantiated, and simulated their own process models

# Course Schedule

Day 1

9:00 Course Introduction and Overview
- SPMS Overview
- Process Models vs Project Plans
- Granularity Issues
- Nodes, Constraints, Phases
- Architectural Levels
- Development Modes
- Hierarchical Models
- Measurement Model
- Graphical Representation

10:00 Building a simple Model in Xpert
- Exercise 1

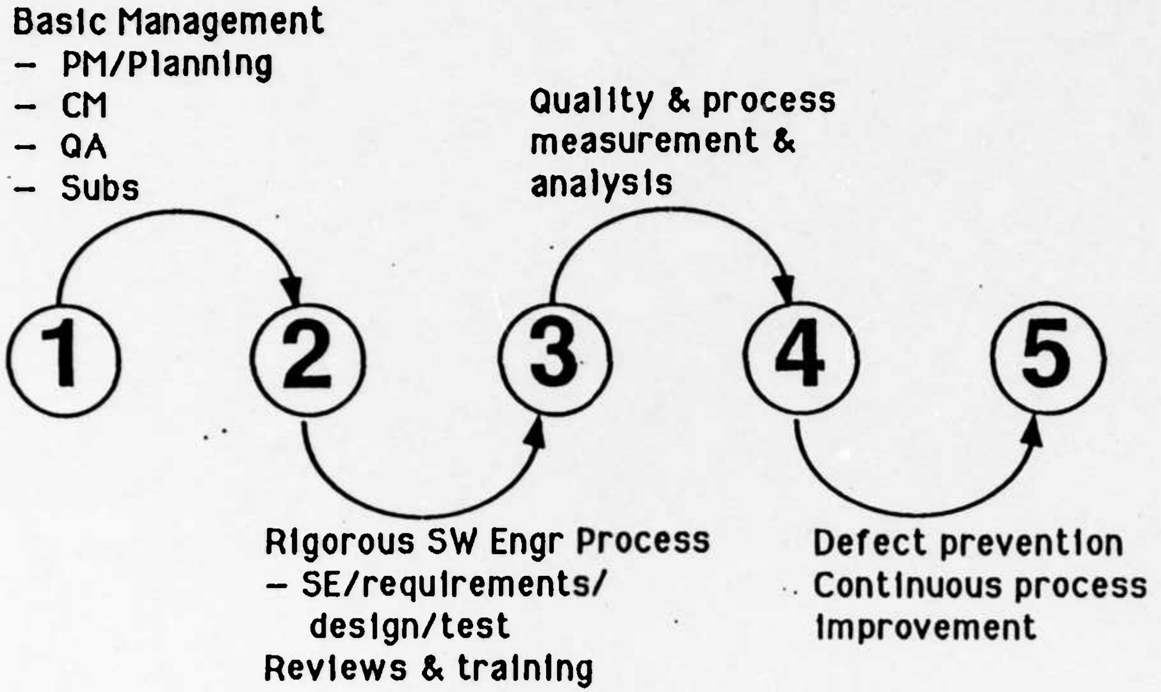
10:30 SPMS model representation
- Exercise 2

11:00 Project Specific Data
11:30 Exercise 3
12:00 Lunch
1:00 Discussion of exercises
1:30 Scheduling the Plan
2:00 Exercise 4
2:30 Graphic Monitoring
3:00 Simulation of the Plan
3:30 Exercise 5
4:00 Discussion and review

Day 2

9:00 Review/Questions
9:30 Validation tasks and Rework
10:00 More Complex Models
10:30 Exercise 6
11:00 Discussion & Questions
11:30 Summary and support issues.

# Process Capability Improvement Focus



Basic Management
- PM/Planning
- CM
- QA
- Subs

Quality & process measurement & analysis

Rigorous SW Engr Process
- SE/requirements/ design/test

Reviews & training

Defect prevention
Continuous process improvement

# What is Software Process Management?

Process Model = How

Project Data = What

Plan = How + What

Resources = Who

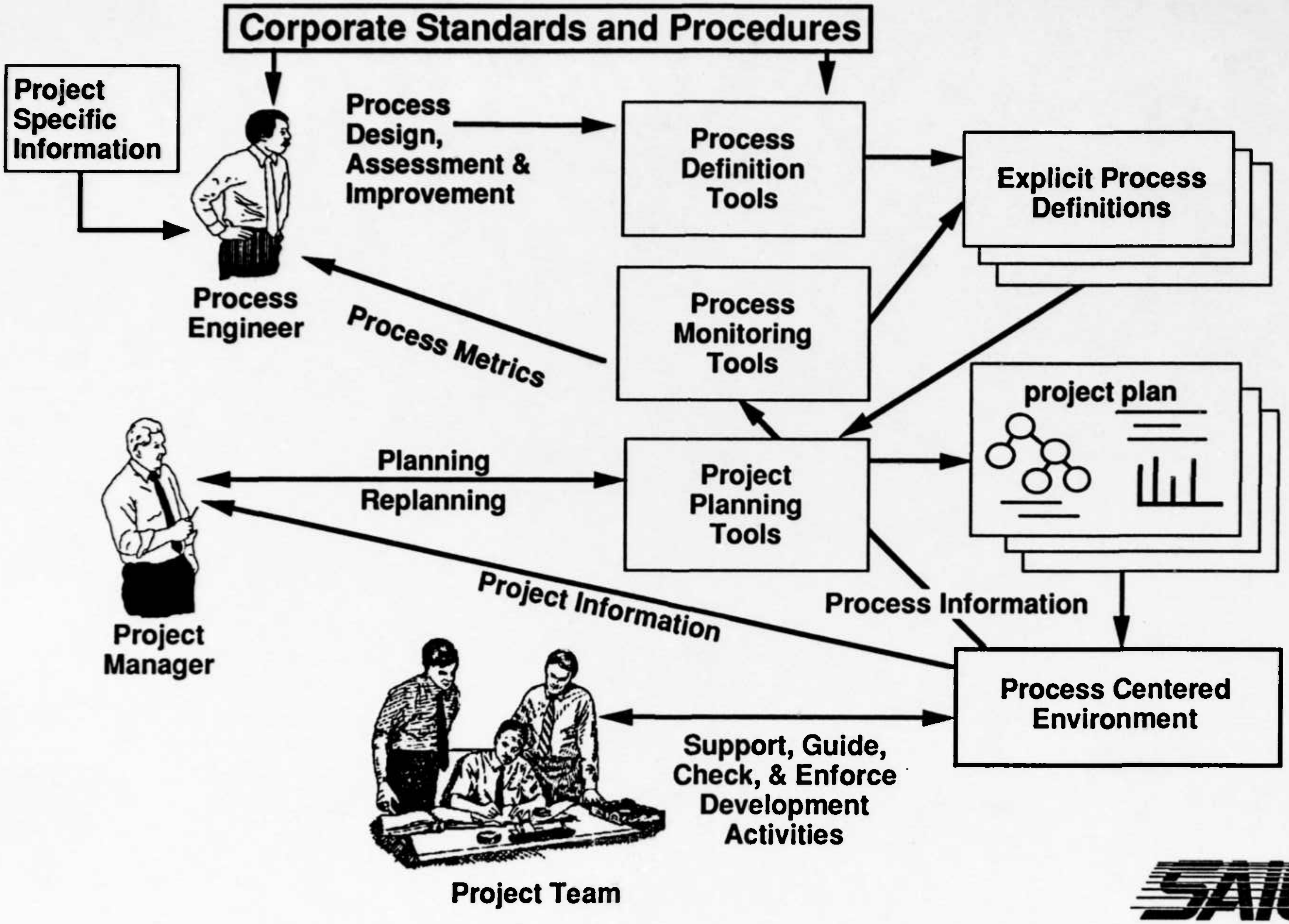
Durations + Scheduling = When

Scheduled Plan with Resources = Who, What, When, How

This Plan + monitoring methods + enaction =

## Software Process Management

# Process Management Paradigm



Corporate Standards and Procedures

Project Specific Information

Process Engineer

Process Design, Assessment & Improvement

Process Definition Tools

Explicit Process Definitions

Process Metrics

Process Monitoring Tools

project plan

Project Manager

Planning

Replanning

Project Planning Tools

Project Information

Process Information

Process Centered Environment

Support, Guide, Check, & Enforce Development Activities

Project Team

# Process Models vs Instantiated Plans

## Process Model:

Represents a prototypical sequence of tasks, milestones, constraints, and products necessary to produce a prototypical single instance of each of the types of software component within the process model.

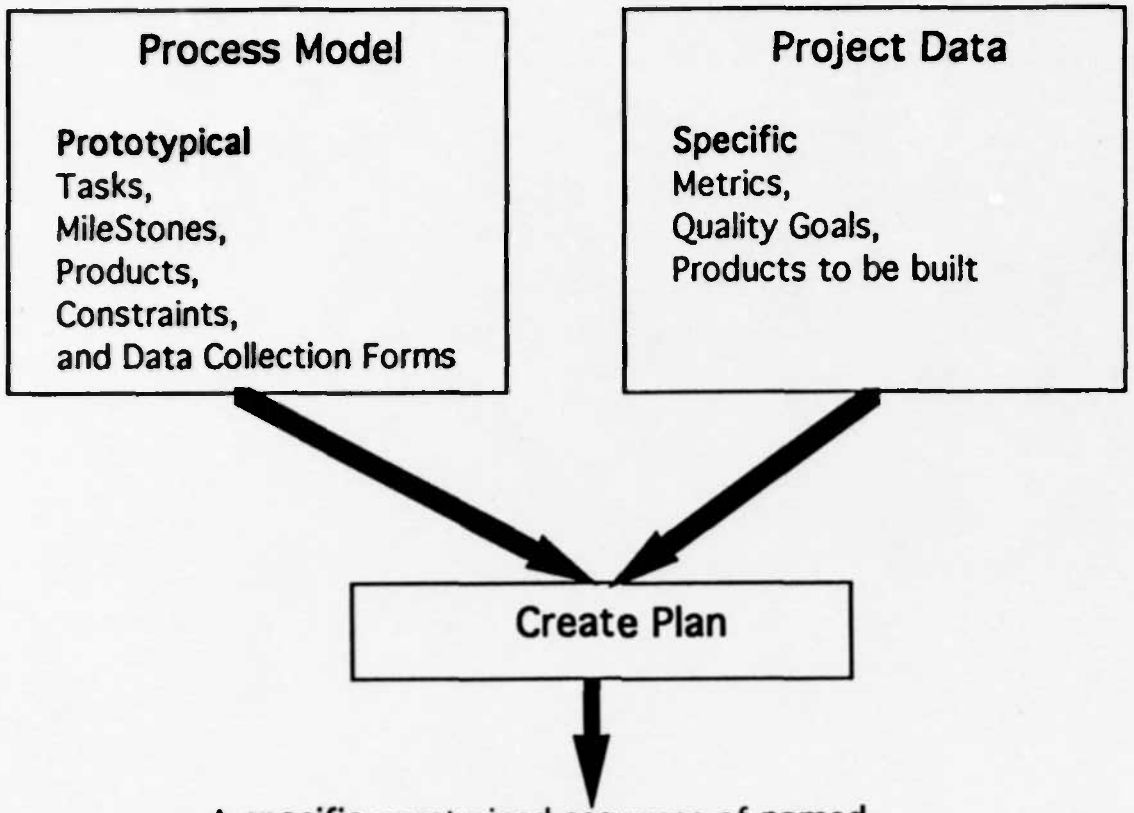
Process models provide the framework for producing plans that may be replicated and a framework for metrics which may measure the process.

## Plans:

Contain numerous specific named and inter-related instances of the software components and the tasks necessary to produce them according to the process model.

Plans are the baselines for monitoring progress of a specific software development project.

# Model + Project Data = Plan



Process Model

**Prototypical**
Tasks,
MileStones,
Products,
Constraints,
and Data Collection Forms

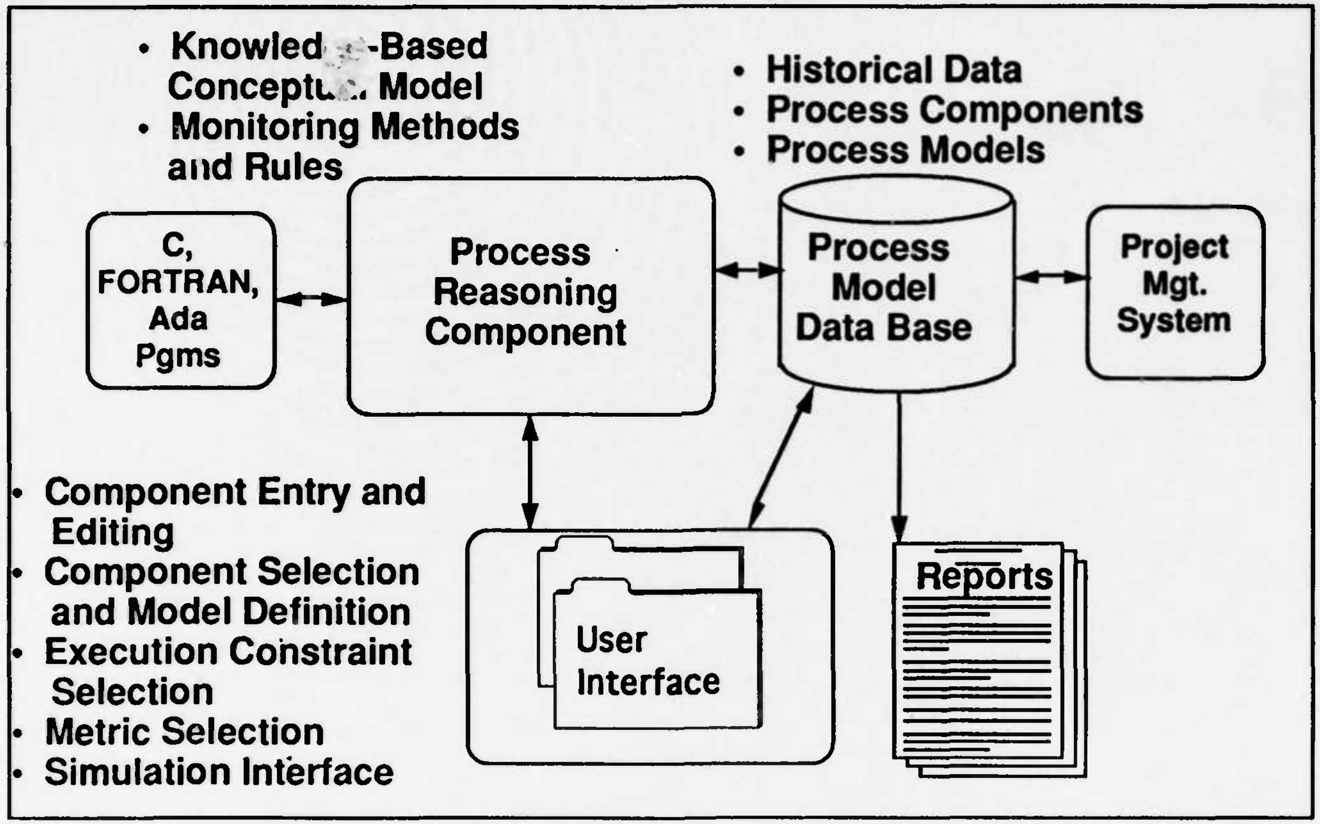
Project Data

**Specific**
Metrics,
Quality Goals,
Products to be built

Create Plan

A **specific** constrained sequence of **named**
Tasks,
MileStones,
Products,
Monitoring methods,
and tables to hold collected metric information.

# Architecture of SPMS



- Knowledge-Based Conceptual Model
- Monitoring Methods and Rules

- Historical Data
- Process Components
- Process Models

C, FORTRAN, Ada Pgms

Process Reasoning Component

Process Model Data Base

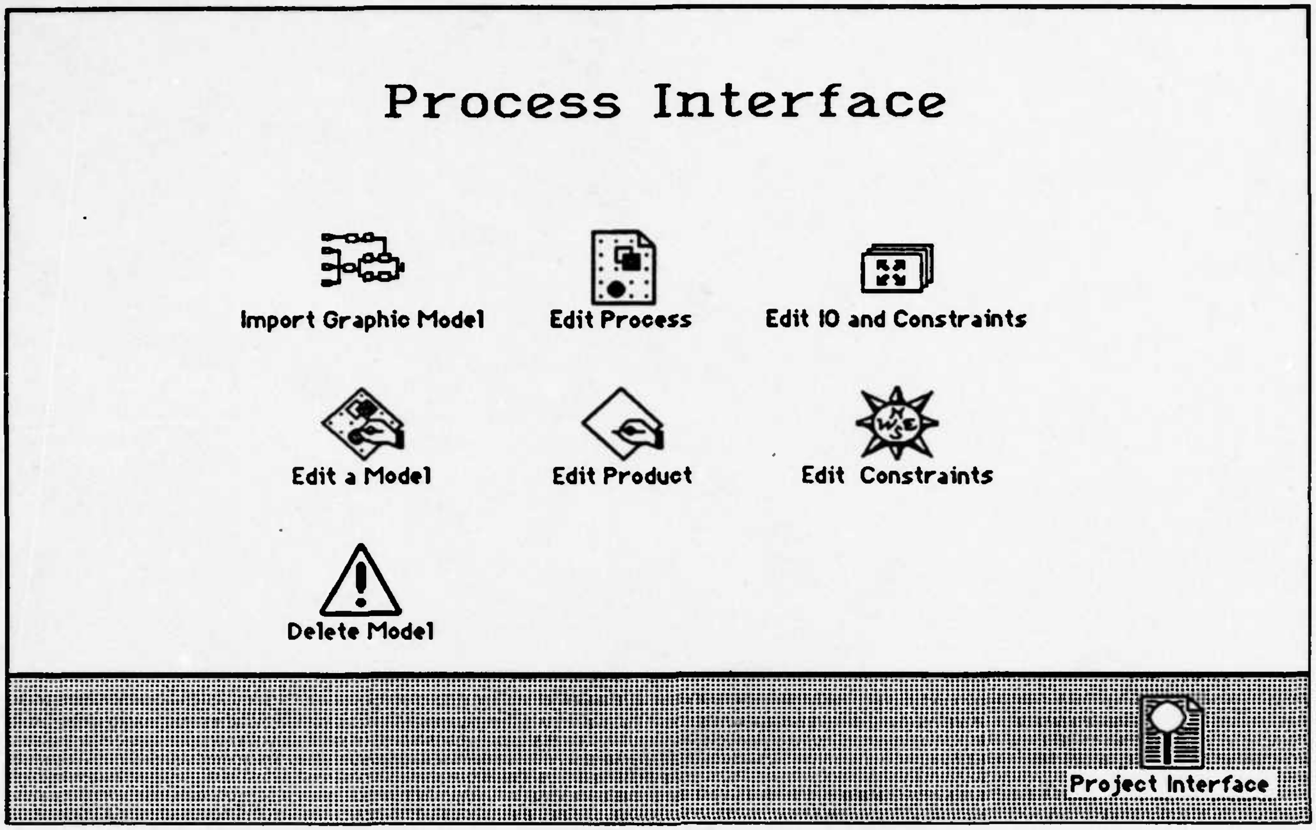
Project Mgt. System

- Component Entry and Editing
- Component Selection and Model Definition
- Execution Constraint Selection
- Metric Selection
- Simulation Interface

User Interface

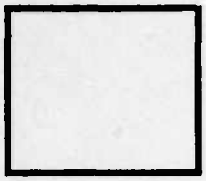
Reports

# Process Model Granularity Issues



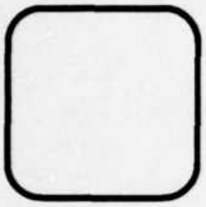
- Entire model
  - Import Graphic Model
  - Delete Model
- Named Groups of Components
  - Edit Model
- Individual Components
  - Edit Process
  - Edit Product
  - Edit Constraints
  - Edit IO and Constraints

# Nodes



- Task:

The basic component of a process model.



- Milestone:

A special node used to highlight important events in a process model.

- Interface:

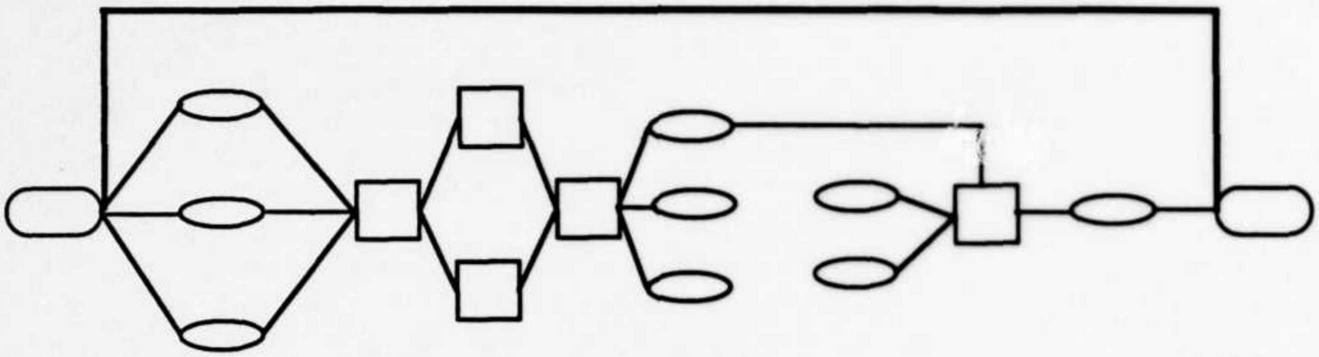
A special node used to link subnetworks in the process model. Represents a product in the SPMS.

- Reverse: ( OR )

A special node used which allows the successor to start when *any* predecessor is complete rather than when *all*predecessors are complete.

# Constraints

Systems Analysis Phase



- Finish to Start:

  - Specifies that a task cannot start until its predecessor is complete.

- Finish to Finish:

  - Specifies that the completion of a task dictates the completion of its successor. May have a duration to specify lag time between a task and its successor.

- Start to Finish:

  - Specifies that a task cannot finish until its predecessor starts.

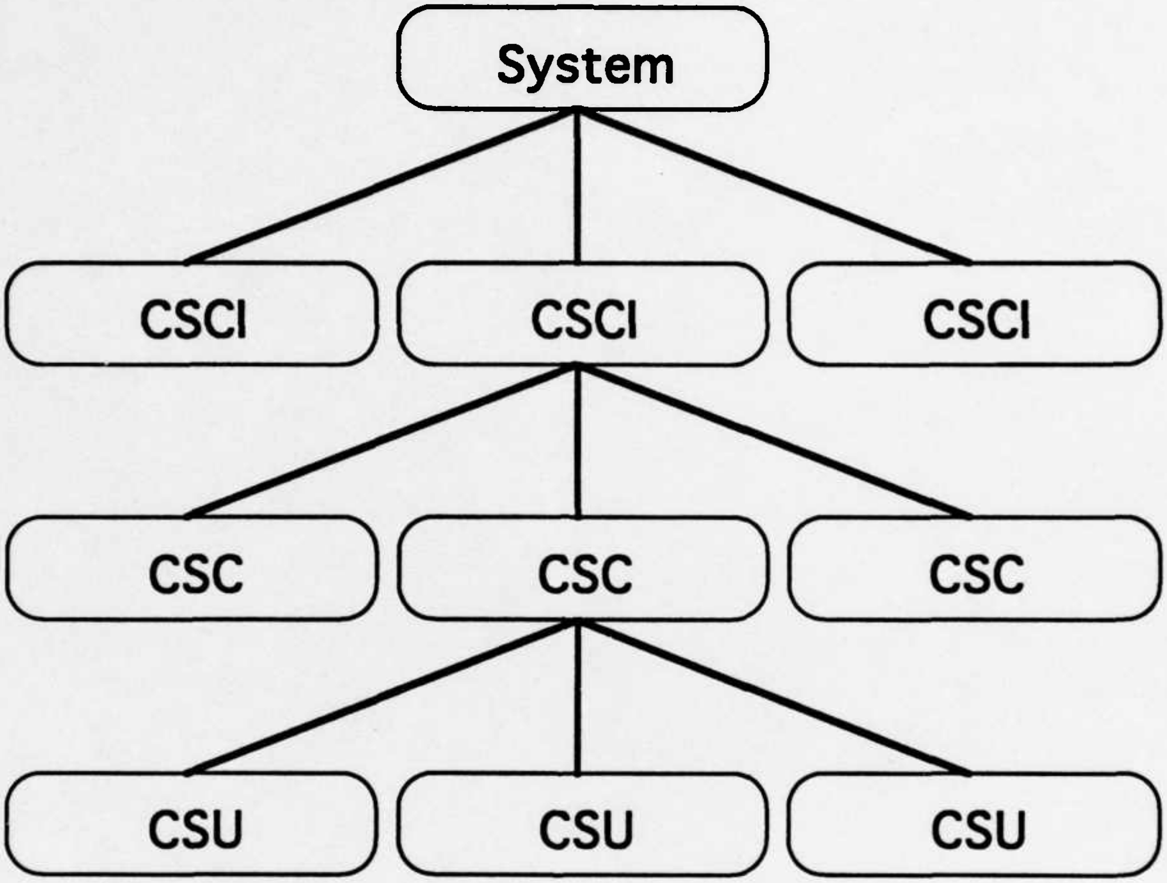
- Start to Start:

  - Specifies that two tasks can start together. A duration applied to this type of constraint indicates a lead time between the start of one task and the start of its predecessor.

- Hammock links:

  - May have resources
  - Dynamic durations calculated as elapsed time between end points.
  - Used to represent phases in SPMS

# Architectural Levels



- Alternative architectural levels within a process model
- Nodes in a process model contain an architectural level parameter.
- Project specific software components also contain this parameter.
- Plans contain nodes in which the model and component parameters match on both architectural level and development mode.

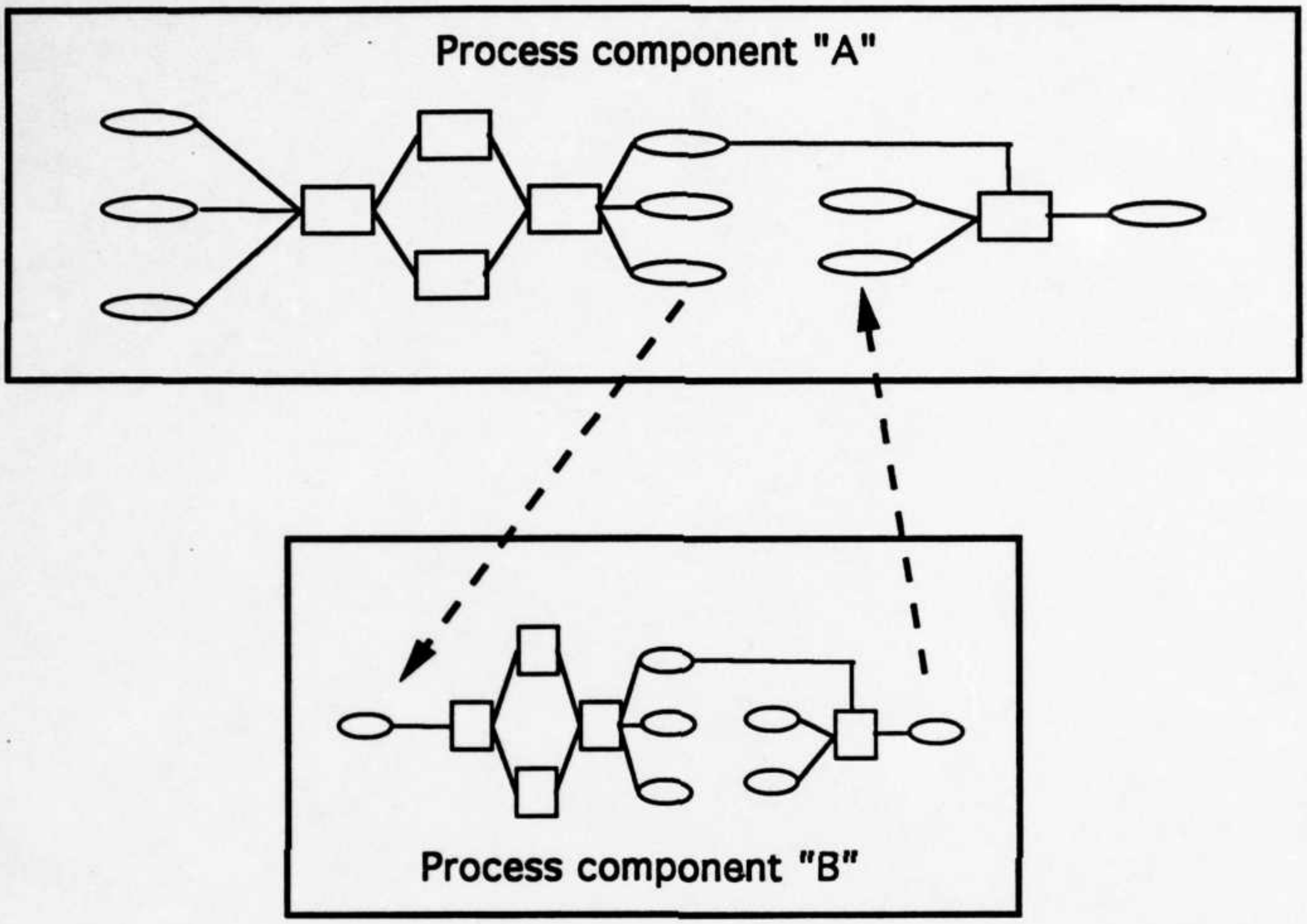
SAIC®

# Development Modes

- Develop module?
- Reuse module?
- Prototype module?
- User defined....

- Alternative sequences of nodes within a process model
- Nodes in a process model contain a development mode parameter.
- Project specific software components also contain this parameter.
- Plans contain nodes in which the model and component parameters match on both architectural level and development mode.
- Allows instantiation time tailoring of model.

# Hierarchical Models



Some Possibilities:

Process component "A" might be at the SYSTEM level.
Process component "B" might be at the CSCI level.
"B" might be considered "Part of" "A"

Process components "A" and "B" might be at the same level and "B" specify greater detail than other portions of "A". "B" might be considered "Part of" "A"

How do you want to view parts in the model editor?

# Executable Process Models and Metrics

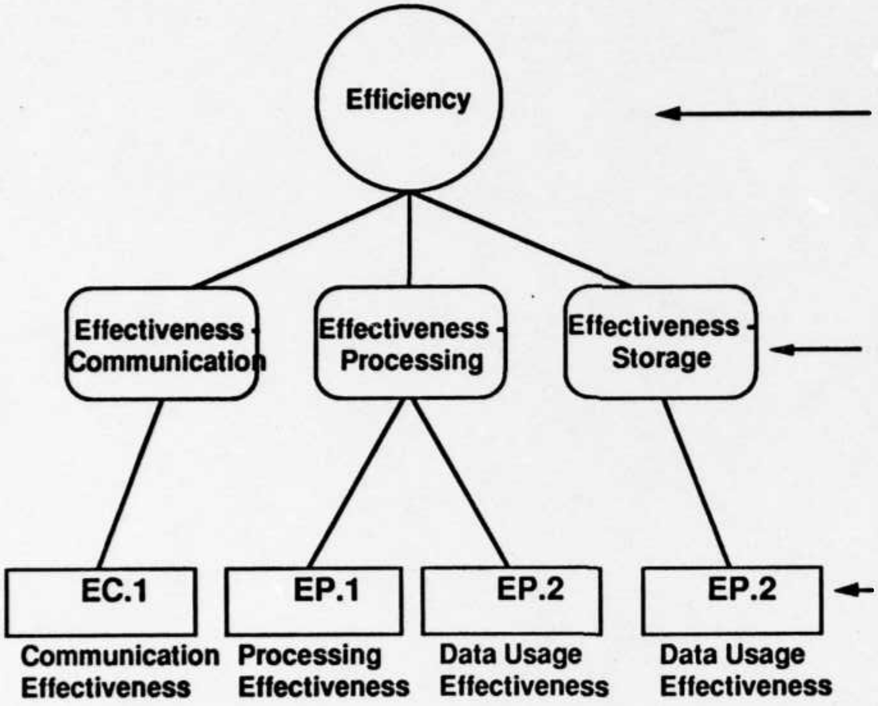
"Define the process model, then the metrics.You must know what you are doing before measuring how well you do it...The need to collect metrics on a process and items it produces... is defined in the exit criteria for that process. Thus, the process model for data collection and analysis becomes integral to the production process model ...An important secondary discovery about the interaction of metrics and process models was that metric collection could drive model refinement." (N. Ross, July'90)

Without metrics the exit criteria for executable process components cannot be evaluated

Without process models one cannot effectively identify what or when to measure, or know what actions should be taken as a result of metric evaluation

Both metrics and process models are integral to SPMS's

# RADC Quality Framework



Efficiency

Effectiveness - Communication

Effectiveness - Processing

Effectiveness - Storage

EC.1 Communication Effectiveness

EP.1 Processing Effectiveness

EP.2 Data Usage Effectiveness

EP.2 Data Usage Effectiveness

Factors are management oriented views of Product Quality.

Criteria are software oriented attributes which provide quality.

Metrics are quantitative measures of those attributes.

# TABLE 1.1-3 SOFTWARE QUALITY FRAMEWORK FACTORS AND ASSOCIATED CRITERIA

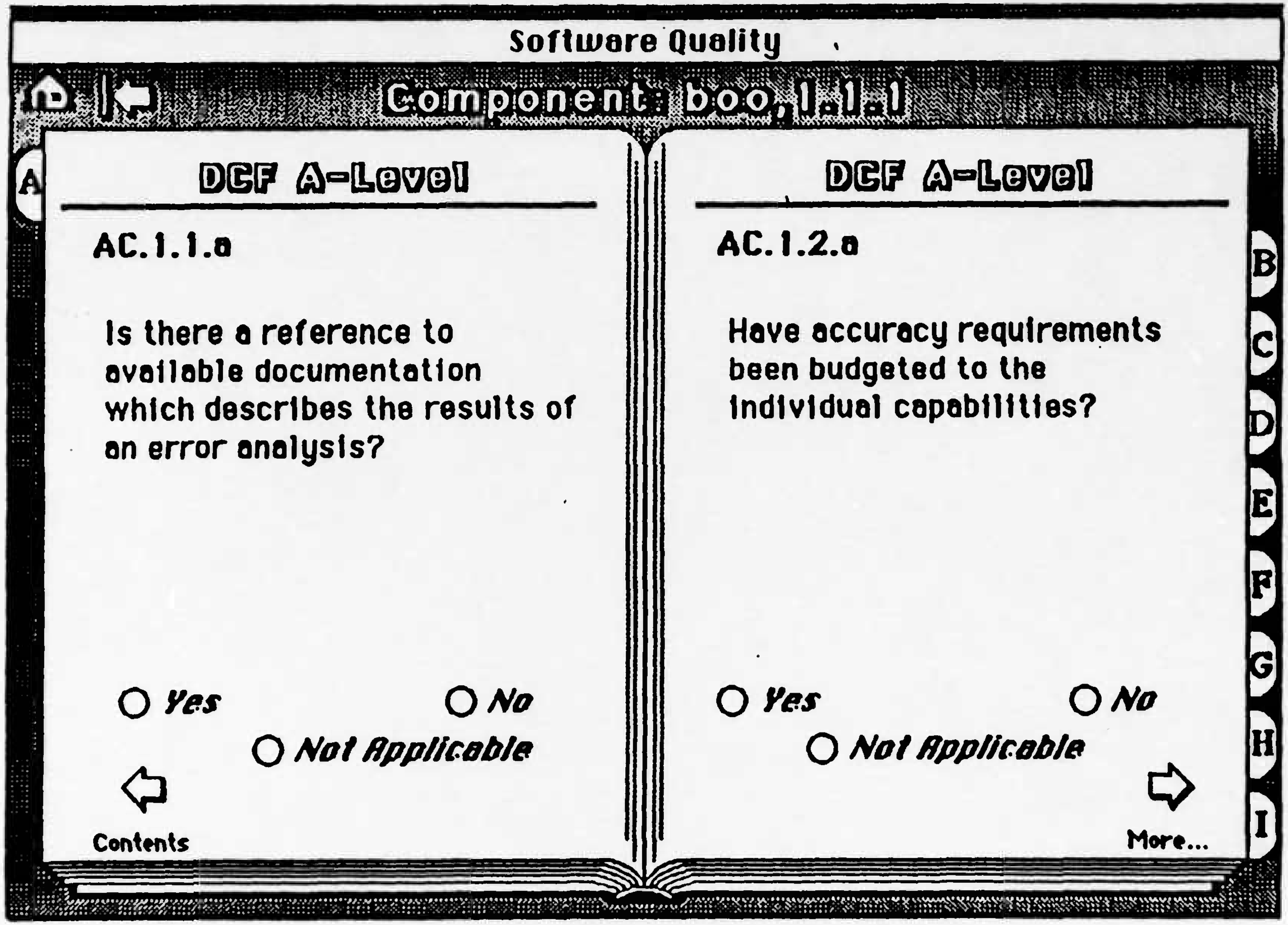
| FACTOR / CRITERION | EFFICIENCY | INTEGRITY | RELIABILITY | SURVIVABILITY | USABILITY | CORRECTNESS | MAINTAINABILITY | VERIFIABILITY | EXPANDABILITY | FLEXIBILITY | INTEROPERABILITY | PORTABILITY | REUSABILITY |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ACCURACY | | | X | | | | | | | | | | |
| ANOMALY MANAGEMENT | | | X | X | | | | | | | | | |
| AUTONOMY | | | | X | | | | | | | | | |
| DISTRIBUTEDNESS | | | | X | | | | | | | | | |
| EFFECTIVENESS - COMMUNICATION | X | | | | | | | | | | | | |
| EFFECTIVENESS - PROCESSING | X | | | | | | | | | | | | |
| EFFECTIVENESS - STORAGE | X | | | | | | | | | | | | |
| OPERABILITY | | | | | X | | | | | | | | |
| RECONFIGURABILITY | | | | X | | | | | | | | | |
| SYSTEM ACCESSIBILITY | | X | | | | | | | | | | | |
| TRAINING | | | | | X | | | | | | | | |
| COMPLETENESS | | | | | | X | | | | | | | |
| CONSISTENCY | | | | | | X | X | | | | | | |
| TRACEABILITY | | | | | | X | | | | | | | |
| VISIBILITY | | | | | | | X | X | | | | | |
| APPLICATION INDEPENDENCE | | | | | | | | | | | | | X |
| AUGMENTABILITY | | | | | | | | | X | | | | |
| COMMONALITY | | | | | | | | | | | X | | |
| DOCUMENT ACCESSIBILITY | | | | | | | | | | | | | X |
| FUNCTIONAL OVERLAP | | | | | | | | | | | X | | |
| FUNCTIONAL SCOPE | | | | | | | | | | | | | X |
| GENERALITY | | | | | | | | | X | X | | | X |
| INDEPENDENCE | | | | | | | | | | | X | X | X |
| SYSTEM CLARITY | | | | | | | | | | | | | X |
| SYSTEM COMPATIBILITY | | | | | | | | | | | X | | |
| VIRTUALITY | | | | | | | | | X | | | | |
| MODULARITY | | | | X | | | X | X | X | X | X | X | X |
| SELF-DESCRIPTIVENESS | | | | | | | X | X | X | X | | X | X |
| SIMPLICITY | | | X | | | | X | X | X | X | | | X |

# Data Collection Forms by Phase and Architectural Level

| Software Development Phase | Form |
|---|---|
| System requirements analysis/design | DCF A |
| Software requirements analysis | DCF B |
| Preliminary design | DCF C |
| Detailed design | DCF D |
| Coding and CSU testing | DCF E |
| CSC integration and test | DCF F |
| CSCI testing | DCF G |
| System testing | DCF H |
| Operational test and evaluation | DCF I |

| Architectural Level | Data Collection Forms | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G | H | I |
| SYSTEM | X | | | | | | | X | X |
| CSCI | | X | X | X | X | X | X | | |
| CSC | | | X | X | X | X | | | |
| CSU | | | | X | X | X | | | |

SAIC®



Software Quality

Component: boo,1.1.1

DCF A-Level

AC.1.1.a

Is there a reference to available documentation which describes the results of an error analysis?

○ Yes ○ No
○ Not Applicable

DCF A-Level

AC.1.2.a

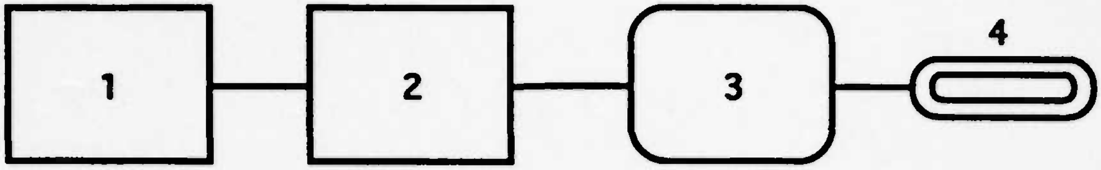
Have accuracy requirements been budgeted to the individual capabilities?

○ Yes ○ No
○ Not Applicable

## Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

| METRIC | DCF | LEVEL | METRIC ELEMENTS FORMULA |
|---|---|---|---|
|  | b | CSCI | (SI.2.1.b) |
|  | c | CSCI | (SI.2.1.c) |
|  | d | CSU | (SI.2.1.d) |
|  | g | CSCI | (SI.2.1.g) |
|  | h | System | (SI.2.1.h) |
| SI.3 | d | CSU | 1 / (1 + SI.3.1.d + SI.3.2.d) |
|  | e | CSU | 1 / (1 + SI.3.1.e + SI.3.2.e) |
| SI.4 | d | CSU | a*(SI.4.1.d) + b*(1 - (SI.4.3.d / AU.1.2.d)) + c*(1 - (SI.4.5.d / SI.4.6.d)) + d*(1 - (SI.4.7.d / SI.4.6.d)) + e*(SI.4.8.d) + f*(1 / SI.4.10.d) + g*(1 - (SI.4.11.d / AU.1.2.d)) + h*(1 - ( (SI.4.12.d + SI.4.13.d) / AU.1.2.d)) + i*(SI.4.15.d / SI.4.14.d) + j*(SI.4.16.d) |
|  | d | CSCI | (SI.4.18.d) |
|  | e | CSU | a*(SI.4.1.e) + b*(SI.4.2.e) + c*(1 / (1 + SI.4.3.e)) + d*(1 - (SI.4.5.e / SI.4.6.e)) + e*(1 - (SI.4.7.e / SI.4.6.e)) + f*(SI.4.8.e) + g*(1 - (SI.4.9.e / AP.3.3.e)) + h*(1 / SI.4.10.e) + i*(1 - (SI.4.11.e / AP.3.3.e)) + j*(1 - (SI.4.12.e + SI.4.13.e) / AP.3.3.e) + k*(SI.4.15.e / SI.4.14.e) + l*(SI.4.16.e) + m*(SI.4.17.e) |
|  | e | CSCI | (SI.4.18.e) |
| SI.5 | d | CSU | a*(1 / (1 + SI.5.1.d)) + b*(SI.5.3.d / SI.5.2.d) + c*(SI.5.4.d) |
|  | e | CSU | a*(1 / (1 + SI.5.1.e)) + b*(SI.5.3.e / SI.5.2.e) + c*(SI.5.4.e) |

# Validation Tasks and Rework Node Id'S



Task "A"
Process id
123

Task "B"
DCF A, 123
SYSTEM

Milestone
"C"

Product
"D"

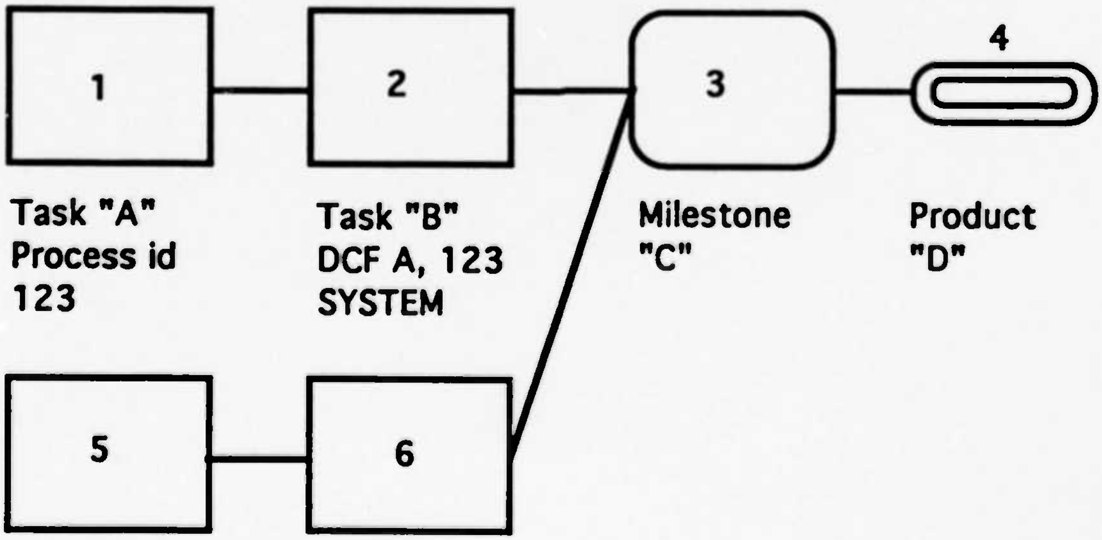
Task "B" is a validation task.

When Task "B" begins it requests that the Data Collection Form (DCF A) associated with it be filled with data.

When the data becomes available, the metrics associated with this project are computed and compared with the quality goals of Product "D".

- Quality goals met, then Task "B" is complete and the Milestone "C" is met and Product "D" becomes available.

- Quality goals not met then replanning will find the task associated with product "D" which is of the process type "123" and use it as the starting point for cloning up to and including the validation task.



Task "A"
Process id
123

Task "B"
DCF A, 123
SYSTEM

Milestone
"C"

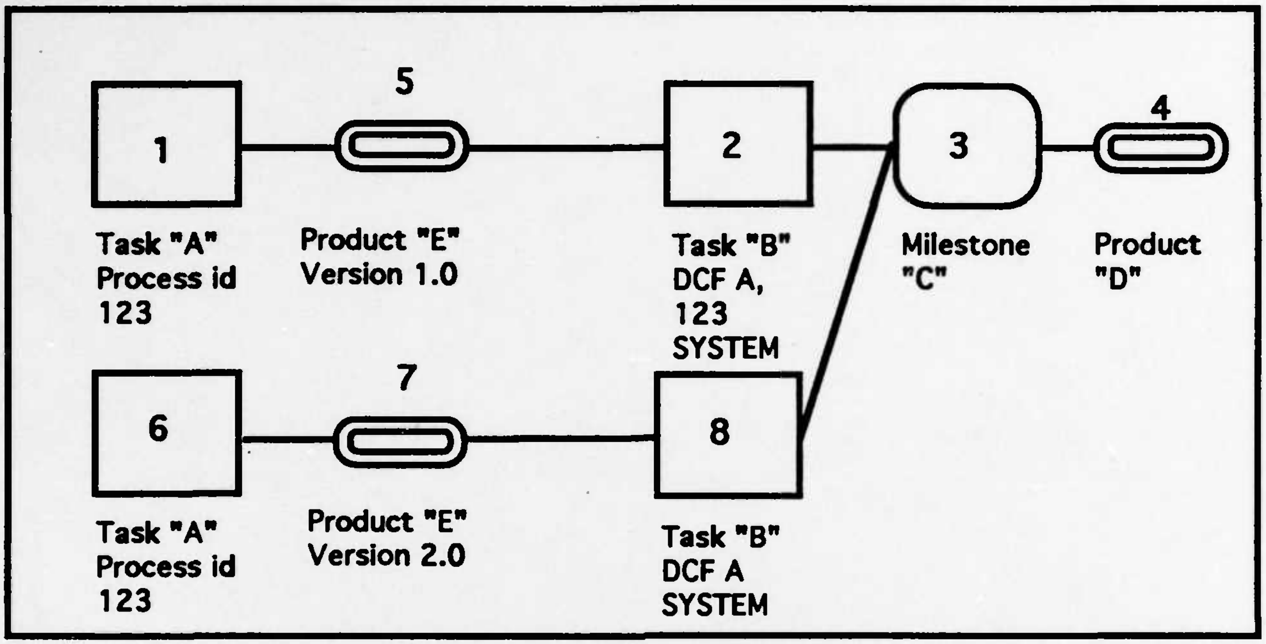
Product
"D"

Task "A"
Process id
123

Task "B"
DCF A
SYSTEM

# Impact of Re-Work on other Process Components
# Tasks Not Started Case

Process Component 1



1
Task "A"
Process id
123

5
Product "E"
Version 1.0

2
Task "B"
DCF A,
123
SYSTEM

3
Milestone
"C"

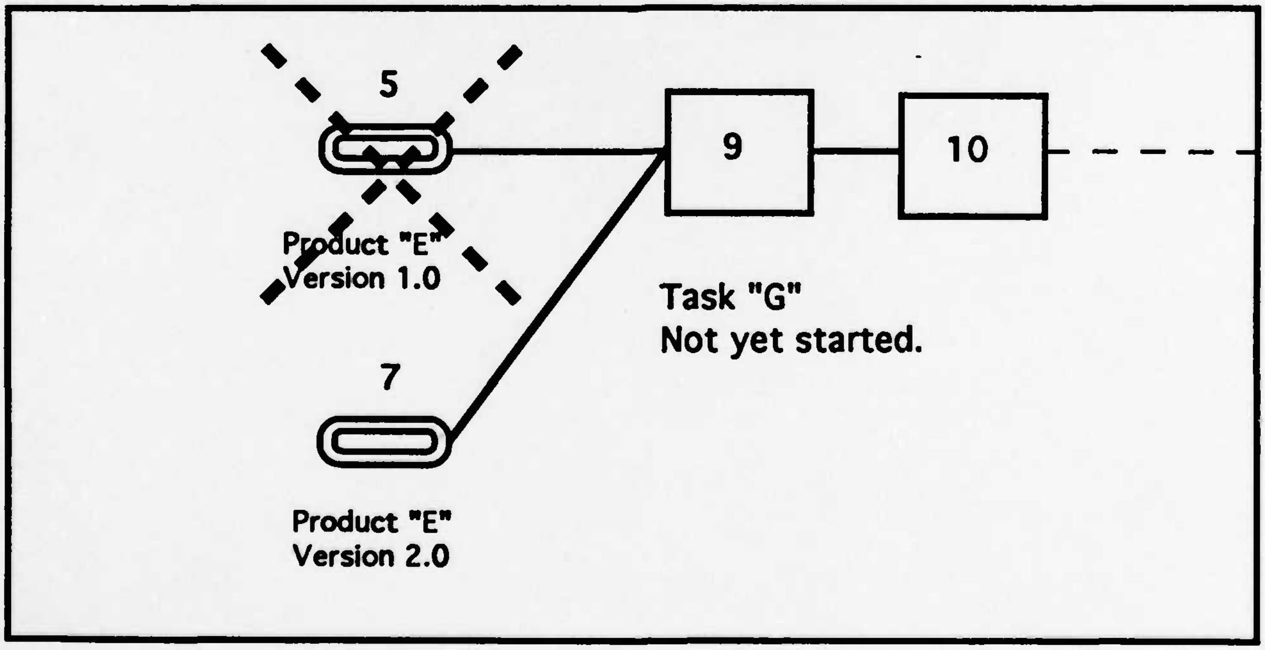
4
Product
"D"

6
Task "A"
Process id
123

7
Product "E"
Version 2.0

8
Task "B"
DCF A
SYSTEM

Process Component 2



5
Product "E"
Version 1.0

7
Product "E"
Version 2.0

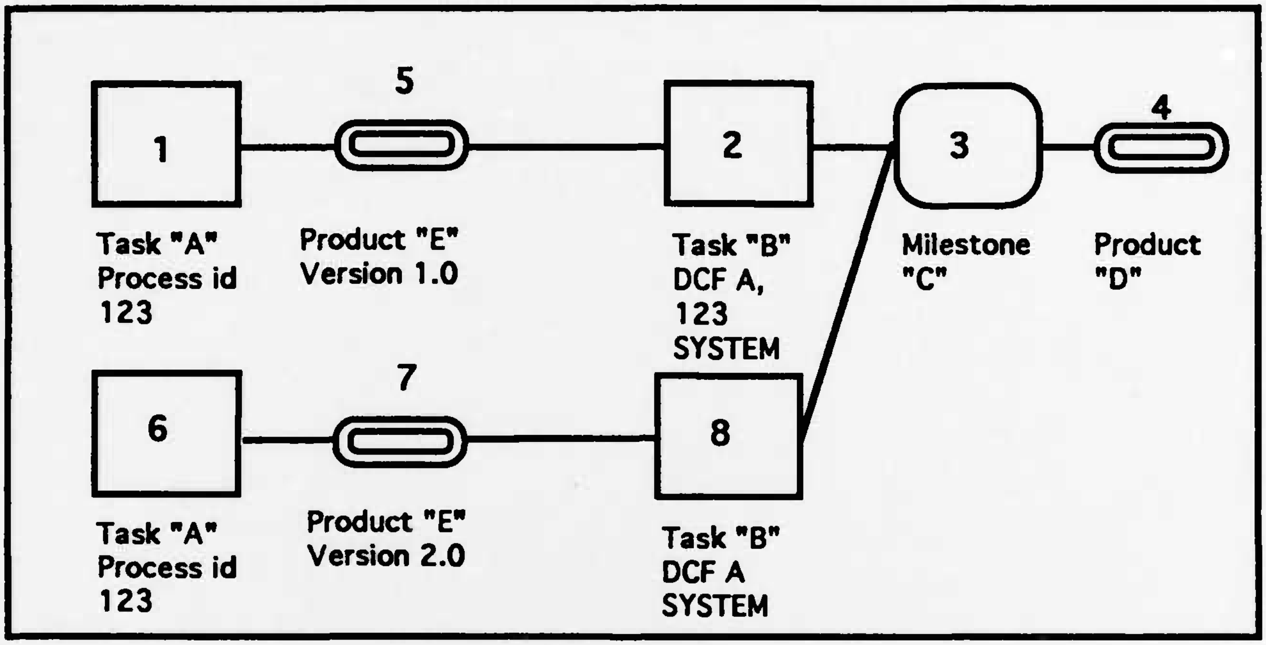
9

10

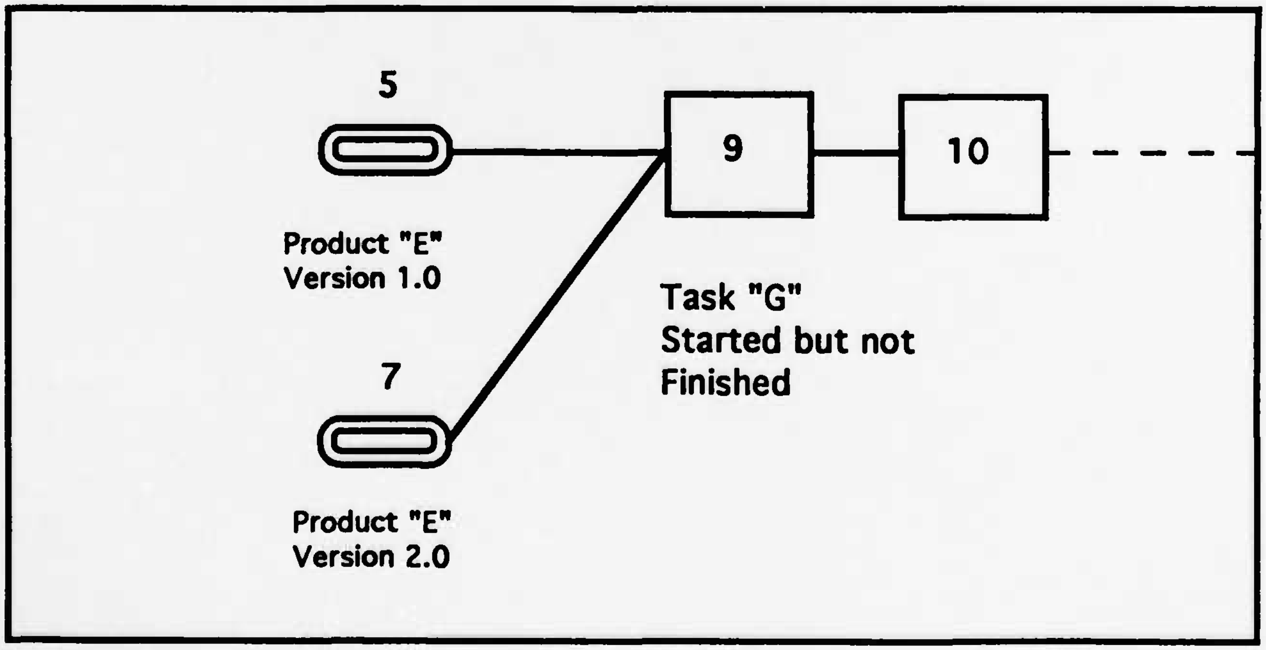
Task "G"
Not yet started.

# Impact of Re-Work on other Process Components
# Tasks In Progress Case

Process Component 1

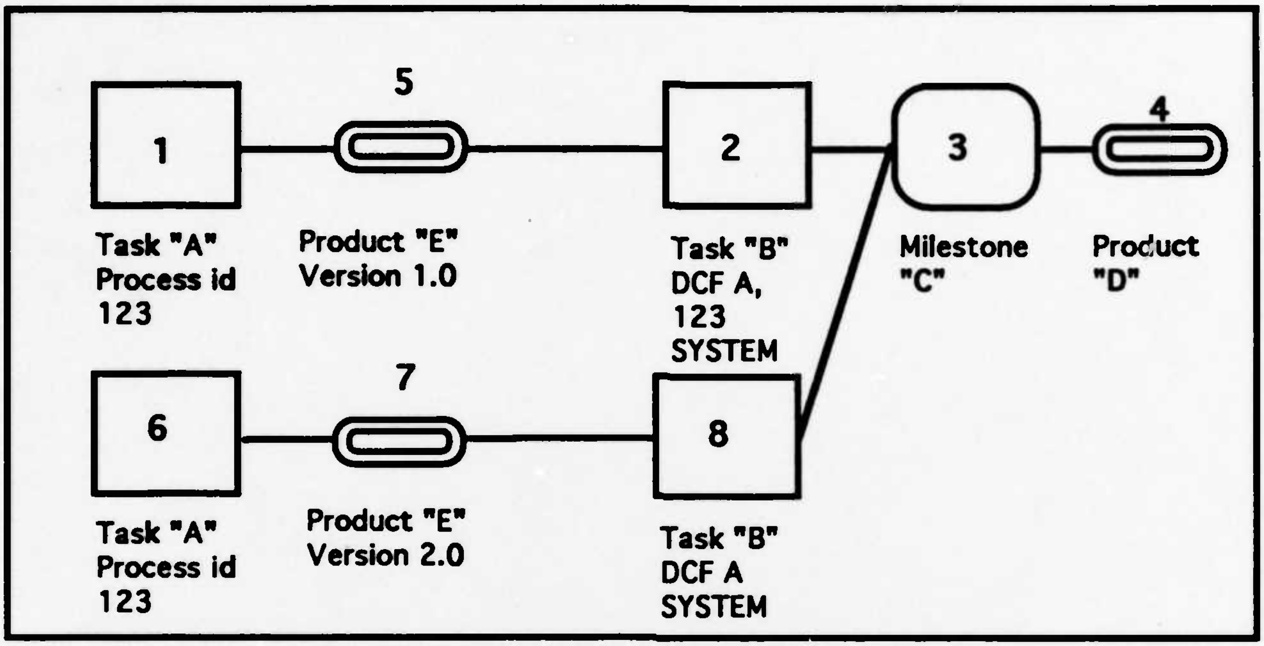


Process Component 2



# Impact of Re-Work on other Process Components
# Tasks Complete Case

Process Component 1



5

1

2

3

4

Task "A"
Process id
123

Product "E"
Version 1.0

Task "B"
DCF A,
123
SYSTEM

Milestone
"C"

Product
"D"

7

6

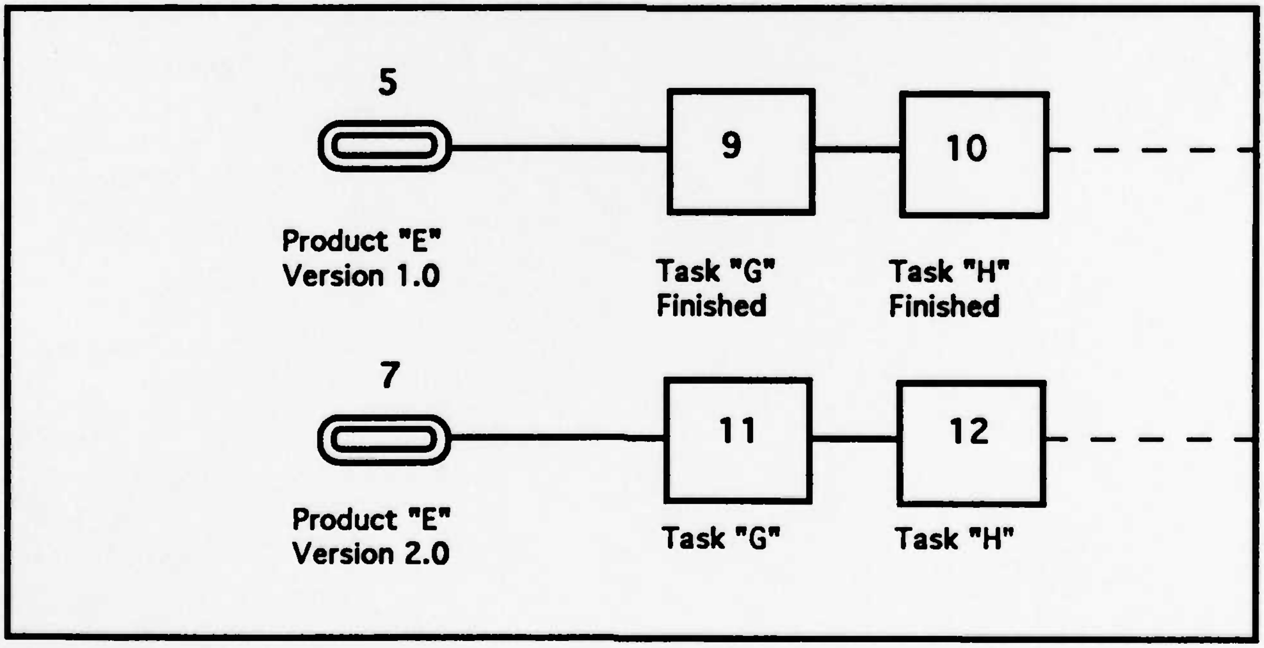
8

Task "A"
Process id
123

Product "E"
Version 2.0

Task "B"
DCF A
SYSTEM

Process Component 2



5

9

10

Product "E"
Version 1.0

Task "G"
Finished

Task "H"
Finished

7

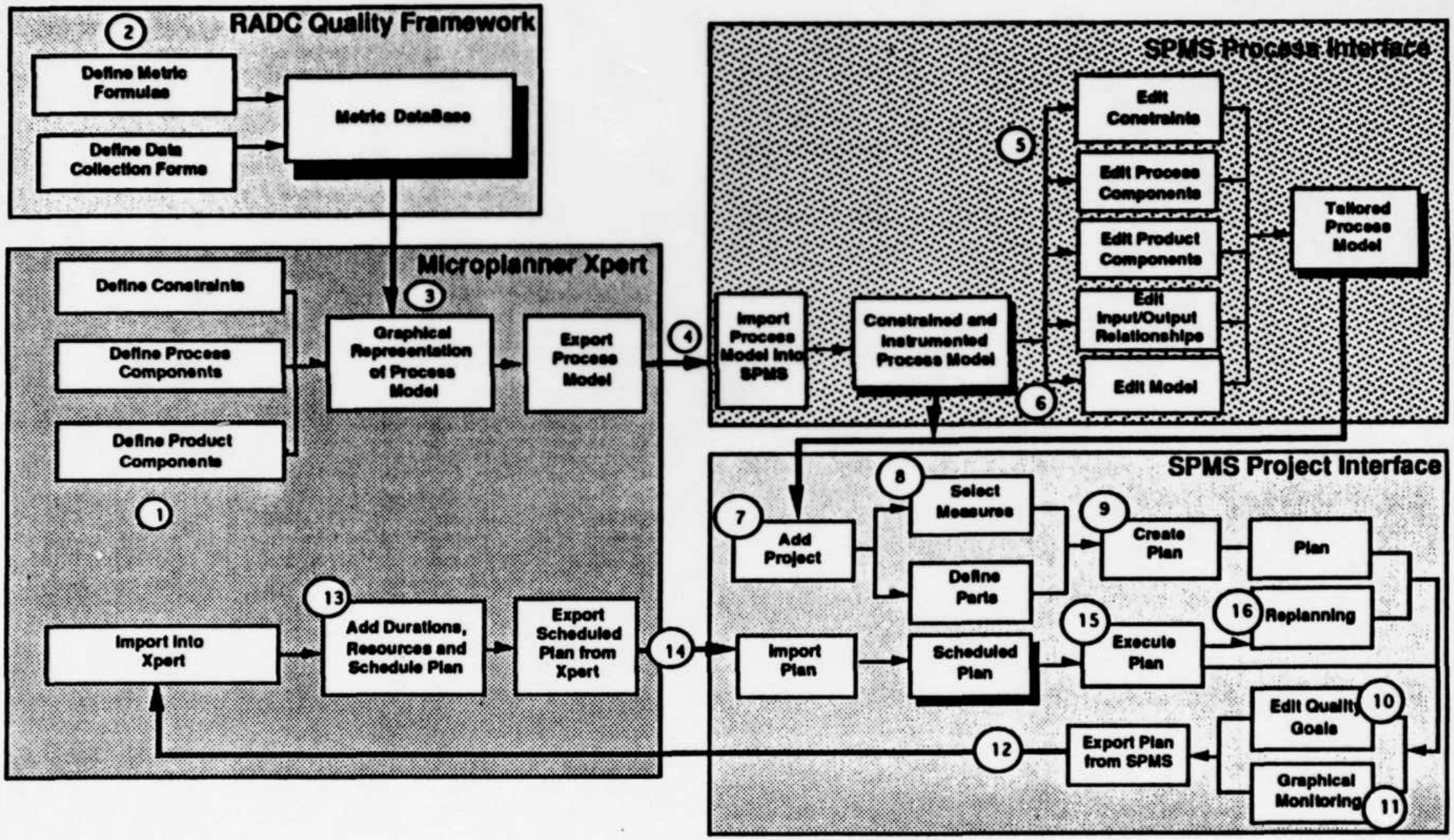
11

12

Product "E"
Version 2.0

Task "G"

Task "H"


RADC Quality Framework
2
Define Metric Formulae
Define Data Collection Forms
Metric DataBase
Microplanner Xpert
Define Constraints
Define Process Components
Define Product Components
1
3
Graphical Representation of Process Model
Export Process Model
4
13
Import into Xpert
Add Durations, Resources and Schedule Plan
Export Scheduled Plan from Xpert
14
SPMS Process Interface
Import Process Model into SPMS
Constrained and Instrumented Process Model
5
6
Edit Constraints
Edit Process Components
Edit Product Components
Edit Input/Output Relationships
Edit Model
Tailored Process Model
SPMS Project Interface
7
Add Project
8
Select Measures
Define Parts
9
Create Plan
Plan
16
Replanning
Import Plan
Scheduled Plan
15
Execute Plan
10
Edit Quality Goals
11
Graphical Monitoring
12
Export Plan from SPMS

# Exércise 1

1. Using the associated hand-out input the System Requirements Analysis subset of the process model as a part of a larger model.

2. Create a Systems Analysis Phase .

3. Use only Finish-to-Start constraints.

4. Export the model from Xpert.

## 6. DEVELOPMENT PROCESS

6.0.1 The developer shall implement a process for developing the software. The development process will be composed of the following major activities:

a. System Requirements Analysis
b. System Design
c. Software Requirements Analysis
d. Software Architectural Design
e. Software Detailed Design
f. Software Coding and Testing
g. Software Integration and Testing
h. Software Configuration Item Testing
i. System Integration and Testing
j. System Testing
k. System Installation and Checkout.

6.0.2 The developer shall select and map these activities onto the life cycle model established for the software project. The selected activities may overlap and may be performed iteratively or recursively.

6.0.3 The developer shall use the methodologies, standards, and procedures that are systematic, adequately documented, and established by the developer's organization for performing the activities.

6.0.4 The developer shall use the computer programming language(s) as specified in the contract to code the deliverable software.

6.0.5 The developer shall consider incorporating non-developmental items into the deliverable software. Incorporation of such items shall comply with the documentation, data rights, warranty, and other requirements as specified in the contract.

6.0.6 The developer may employ non-deliverable items in the development or maintenance of software. However, the developer shall ensure that the operation and maintenance of software after its delivery to the purchaser will be independent of such non-deliverable items. In case such independence cannot be ensured, the developer shall treat the non-deliverable items as non-developmental upon notifying the purchaser regarding the impact of non-deliverable items on the cost, schedule, operation, and maintenance of the deliverable software.

6.1 System Requirements Analysis. The developer shall perform or support the following system requirements analysis activities.

6.1.1 Engineering. The developer shall analyze the statement of need, statement of work, and recommended solutions, if available, to define a set of system requirements addressing the following as a minimum:

a. Functions and capabilities of the total system, including performance, quality, and physical characteristics and environmental conditions under which the system will perform;
b. Safety requirements, including those related to equipment characteristics and degradation, methods of operation and maintenance, environmental influences, and personnel injury;
c. Security requirements, including those related to operational and maintenance environments and compromise of sensitive information or materials;
d. Human engineering requirements, including those related to constraints on personnel, areas needing concentrated human attention and sensitive to human errors, and training;
e. Interfaces requirements for interfaces external to the system, including interfaces with users;
f. Operation and maintenance requirements.

6.1.2 Qualification Testing. The developer shall define a set of system qualification requirements, including qualifications methods, for verification, validation, and testing of the system requirements.

6.1.3 Documentation. The developer shall document the system and qualification requirements in a system requirements document in accordance with section 9.4.

6.1.4 Product Evaluation. The developer shall perform evaluations of the system and qualification requirements for the criteria identified below as a minimum. When a problem is detected, corrective actions shall be taken in accordance with section 9.3.3.

a. Consistency -- external and internal;
b. Traceability;
c. Test coverage of requirements;
d. Testability;
e. Feasibility of design, operation, and maintenance;

6.1.5 Formal Review. The developer shall conduct one or more system requirements reviews in accordance with section 9.2.

6.1.6 Configuration Management. The developer shall place the documents identified below under configuration management and perform configuration control in accordance with section 9.1:

a. Statement of work
b. System requirements document.

# Exercise 2

1. Import the graphic model of Exercise 1 into the SPMS.

2. Browse the fine grained components using "Edit Process", "Edit Products", and "Edit Constraints" buttons.

3. Browse the product to producer and consumer relations using the "Edit IO and Constraints" button

4. Browse the hierachical coarse grained components using the "Edit Model" button

5. Edit your model as desired .

# Exercise 3

1. Move to the Project Interface and create a new Project.

2. Define a system level component with a source to match the development mode of your model.

3. Create a Plan by selecting your model and your project and providing a name for your plan.

4. Export the plan from the SPMS

# Exercise 4

1. Create a new project in XPERT. Clear the New Subproject which is automatically created. From the Date Control Panel, Turn off the Show Hours/Minutes option .

2. Import your plan into XPERT.

3. Use "Clean Up" to improve the readability of the activity network.

4. Select the tasks in the network and enter durations. (Format is "weeks,days".

5. Perform Time Analysis.

6. Build a Gannt chart to display your results. See Gannt chart options for tailoring the chart.

7. Build a Table view. Use "Selected Table" with "0" resources displayed

8. Select the entire table and Export it from XPERT. (use name.DAT" format)

# Exercise 5

1. Import the scheduled plan into the SPMS.

2. Randomize the durations of your plan.

3. Create some graphs for monitoring your plan during execution

4. Execute the plan.

# Exercise 6

1. Open your model in Xpert.

2. Add a validation task for the Systems Requirements Document.

3. Be sure your DCF and rework node id are appropriate!

4. Alter the constraints to allow some tasks to begin when *any* of their inputs are available but only finish when *all* of their required inputs have been made available.

5. Export the model from Xpert.

6. Import the model into SPMS

7. Create a new project.

8. Select measures and default quality goals for this project.

9. Define Parts.

10. Create a new plan using your new model and new project.

11. Execute the plan.

12. Replan as necessary.